Masking

e-Yantra Team
Embedded Real-Time Systems Lab
Indian Institute of Technology-Bombay

IIT Bombay
February 18, 2020

Agenda for Discussion

1. Masking
   - Example
   - Need for masking
2. Masking Operators
   - NOT operator
   - Shift operator
   - OR operator
   - AND operator
   - EXOR operator

Need for masking

Need for masking

Let us consider buzzer example to understand the need:

1. Configure PC.3 pin as Output.

   DDRC = 0x08; // 0000 1000

2. To turn ON the buzzer set PC.3 output HIGH

   PORTC = 0x08; // 0000 1000

3. To turn OFF the buzzer set PC.3 output LOW

   PORTC = 0x00; // 0000 0000

Need for masking

Need for masking

1. Sometimes, we need to change the state of one or more pins of the port thereby keeping the rest of the pins unchanged.

Need for masking

1. Sometimes, we need to change the state of one or more pins of the port thereby keeping the rest of the pins unchanged.
2. AVR is not bit addressable. It is only bit accessible.

Need for masking

1. Sometimes, we need to change the state of one or more pins of the port thereby keeping the rest of the pins unchanged.
2. AVR is not bit addressable. It is only bit accessible.
3. No 'address' to a specific bit.

Need for masking

1. Sometimes, we need to change the state of one or more pins of the port thereby keeping the rest of the pins unchanged.
2. AVR is not bit addressable. It is only bit accessible.
3. No 'address' to a specific bit.
4. Use of different masking operators.

Masking Operators

Masking Operators

In general, there are three operators used for masking:

Masking Operators

In general, there are three operators used for masking:

- ✓ OR operator → to SET a particular bit

Masking Operators

In general, there are three operators used for masking:

- OR operator → to SET a particular bit
- AND operator → to RESET a particular bit

Masking Operators

In general, there are three operators used for masking:

- ✓ OR operator → to SET a particular bit
- ✓ AND operator → to RESET a particular bit
- ✓ EXOR operator → to TOGGLE a particular bit

Masking Operators

In general, there are three operators used for masking:

- ✓ OR operator → to SET a particular bit
- ✓ AND operator → to RESET a particular bit
- ✓ EXOR operator → to TOGGLE a particular bit

Masking Operators

In general, there are three operators used for masking:

- ✓ OR operator → to SET a particular bit
- ✓ AND operator → to RESET a particular bit
- ✓ EXOR operator → to TOGGLE a particular bit

Two more operators can be used:

- NOT operator
- Shift operators

NOT Operator

NOT Operator

1. Purpose: To perform negation on all bits.

NOT Operator

1. Purpose: To perform negation on all bits.
2. Symbol: $\backsim$

NOT Operator

1. Purpose: To perform negation on all bits.
2. Symbol: ∽
3. Example:

$A =$

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

NOT Operator

1. Purpose: To perform negation on all bits.
2. Symbol: ∽
3. Example:

A =

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

NOT Operator

1. Purpose: To perform negation on all bits.
2. Symbol: $\backsim$
3. Example:

A =

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

$\backsim A =$

B7	B6	B5	B4	B3	B2	B1	B0
0	1	1	1	1	1	0	0

Shift Operator

Shift Operator

1. Purpose: To shift all bits by specified bit position.

Shift Operator

1. Purpose: To shift all bits by specified bit position.
2. Types: Left Shift and Right Shift

Shift Operator

1. Purpose: To shift all bits by specified bit position.
2. Types: Left Shift and Right Shift
3. Symbol: Left shift (<<) and right shift (>>)

Shift Operator

1. Purpose: To shift all bits by specified bit position.
2. Types: Left Shift and Right Shift
3. Symbol: Left shift (<<) and right shift (>>)
4. Example:

A =

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

Shift Operator

1. Purpose: To shift all bits by specified bit position.
2. Types: Left Shift and Right Shift
3. Symbol: Left shift (<<) and right shift (>>)
4. Example:

A =

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

Shift Operator

1. Purpose: To shift all bits by specified bit position.
2. Types: Left Shift and Right Shift
3. Symbol: Left shift (<<) and right shift (>>)
4. Example:

$A =$

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

$A{<<}2 =$

B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	1	1	0	0

Shift Operator

1. Purpose: To shift all bits by specified bit position.
2. Types: Left Shift and Right Shift
3. Symbol: Left shift (<<) and right shift (>>)
4. Example:

$A =$

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

$A<<2 =$

B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	1	1	0	0

$A>>2 =$

B7	B6	B5	B4	B3	B2	B1	B0
0	0	1	0	0	0	0	0

OR Operator

OR Operator

1. Purpose: To SET particular bit/s.

OR Operator

1. Purpose: To SET particular bit/s.
2. Symbol: |

OR Operator

1. Purpose: To SET particular bit/s.

2. Symbol: |

3. Truth Table:

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Example

Example

1. Example: Setting a bit :

Example

1. Example: Setting a bit :

   a. Consider register has data 0x83 (unknown to us). We want to set 2nd bit of register and keep rest of the data intact.

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

Example

1. Example: Setting a bit :

   a. Consider register has data 0x83 (unknown to us). We want to set 2nd bit of register and keep rest of the data intact.

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

Example

1. Example: Setting a bit :

   a. Consider register has data 0x83 (unknown to us). We want to set 2nd bit of register and keep rest of the data intact.

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

   b. Expected output is:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

Example of Masking

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

Example of Masking

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

OR

B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	1	0	0

Example of Masking

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

OR

B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	1	0	0

Output same as Expected output:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

Example of Masking

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

OR

B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	1	0	0

Output same as Expected output:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

1. register_name = register_name | 0x04;

Example of Masking

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

OR

B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	0	1	0	0

Output same as Expected output:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

1. register_name = register_name | 0x04;
2. register_name |= 0x04;

Example of Masking with Shift Operator

1. register_name |= 0x04;

Example of Masking with Shift Operator

1. register_name |= 0x04;
2. 0x04 can also be written as 1<<2

Example of Masking with Shift Operator

1. register_name |= 0x04;
2. 0x04 can also be written as 1<<2
3. In general, statement can be written as:

   Register_name |= (1 << *pin_no*)

Example of Masking with Shift Operator

1. register_name $|=$ 0x04;
2. 0x04 can also be written as 1<<2
3. In general, statement can be written as:

   Register_name $|= (1 << pin\_no)$
4. For setting multiple bits at once the statement can be written as:

   Register_name $|= ((1 << pin\_no1) \mid (1 << pin\_no2))$

AND Operator

AND Operator

1. Purpose: To RESET particular bit/s.

AND Operator

1. Purpose: To RESET particular bit/s.
2. Symbol: &

AND Operator

1. Purpose: To RESET particular bit/s.
2. Symbol: &
3. Truth Table:

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Example

Example

1. Example: Resetting a bit :

Example

1. Example: Resetting a bit :

   a. Consider register has data 0x87 (unknown to us). We want to reset pin 2 and keep rest of the data intact.

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

Example

1. Example: Resetting a bit :

   a. Consider register has data 0x87 (unknown to us). We want to reset pin 2 and keep rest of the data intact.

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

Example

1. Example: Resetting a bit :

   a. Consider register has data 0x87 (unknown to us). We want to reset pin 2 and keep rest of the data intact.

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

   b. Expected output is:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

Example

1. Example: Resetting a bit :

   a. Consider register has data 0x87 (unknown to us). We want to reset pin 2 and keep rest of the data intact.

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

   b. Expected output is:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

Example

1. Example: Resetting a bit :

   a. Consider register has data 0x87 (unknown to us). We want to reset pin 2 and keep rest of the data intact.

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

   b. Expected output is:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

Example of Masking

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

Example of Masking

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

AND

B7	B6	B5	B4	B3	B2	B1	B0
1	1	1	1	1	0	1	1

Example of Masking

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

AND

B7	B6	B5	B4	B3	B2	B1	B0
1	1	1	1	1	0	1	1

Output same as Expected output:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

Example of Masking

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

AND

B7	B6	B5	B4	B3	B2	B1	B0
1	1	1	1	1	0	1	1

Output same as Expected output:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

1. register_name = register_name & 0xFB;

Example of Masking

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	1	1	1

AND

B7	B6	B5	B4	B3	B2	B1	B0
1	1	1	1	1	0	1	1

Output same as Expected output:

B7	B6	B5	B4	B3	B2	B1	B0
1	0	0	0	0	0	1	1

1. register_name = register_name & 0xFB;
2. register_name &= 0xFB;

Example of Masking with Shift Operator

1. register_name &= 0xFB;

Example of Masking with Shift Operator

1. register_name &= 0xFB;
2. 0xFB can also be written as $\backsim (1 << 2)$

Example of Masking with Shift Operator

1. register_name &= 0xFB;

2. 0xFB can also be written as $\backsim (1 << 2)$

3. In general, statement can be written as:

   Register_name & $= \backsim (1 << \mathit{pin\_no})$

Example of Masking with Shift Operator

1. register_name &= 0xFB;
2. 0xFB can also be written as $\backsim (1 << 2)$
3. In general, statement can be written as:

   Register_name $\& = \backsim (1 << pin\_no)$

4. For resetting multiple bits at once the statement can be written as:

   Register_name $\&= \backsim ((1 << pin\_no1) \mid (1 << pin\_no2))$

Buzzer Example with Masking

Buzzer Example with Masking

1. Configure PC.3 pin as Output.

   DDRC |= (1 << 3);

2. To turn ON the buzzer set PC.3 output HIGH

   PORTC |= (1 << 3);

3. To turn OFF the buzzer set PC.3 output LOW

   PORTC &= ∽ (1 << 3);

EXOR Operator

EXOR Operator

1. Purpose: To TOGGLE particular bit.

EXOR Operator

1. Purpose: To TOGGLE particular bit.
2. Symbol: ^

EXOR Operator

1. Purpose: To TOGGLE particular bit.
2. Symbol: ^
3. Truth Table:

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

EXOR Operator

1. Purpose: To TOGGLE particular bit.
2. Symbol: ^
3. Truth Table:

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

EXOR Operator

1. Purpose: To TOGGLE particular bit.
2. Symbol: ^
3. Truth Table:

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

4. For one bit:
   Register_name ^ = (1 << pin_no)

EXOR Operator

1. Purpose: To TOGGLE particular bit.
2. Symbol: ^
3. Truth Table:

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

4. For one bit:
   Register_name ^ = (1 << pin_no)
5. For toggling multiple bits:
   Register_name ^= ((1 << pin_no1) | (1 << *pin_no*2))

Thank You!